

Application of Information Theory, Lecture 11

Pseudo-Entropy and Pseudorandom Generators

Iftach Haitner

Tel Aviv University.

June 07, 201

Part I

Motivation

Encryption schemes

Definition 1

A pair of algorithms (E, D) is (perfectly correct) encryption scheme, if for any $k \in \{0, 1\}^n$ and $m \in \{0, 1\}^\ell$, it holds that $D(k, E(k, m)) = m$

Encryption schemes

Definition 1

A pair of algorithms (E, D) is (perfectly correct) encryption scheme, if for any $k \in \{0, 1\}^n$ and $m \in \{0, 1\}^\ell$, it holds that $D(k, E(k, m)) = m$

- What security should we ask from such scheme?

Encryption schemes

Definition 1

A pair of algorithms (E, D) is (perfectly correct) encryption scheme, if for any $k \in \{0, 1\}^n$ and $m \in \{0, 1\}^\ell$, it holds that $D(k, E(k, m)) = m$

- ▶ What security should we ask from such scheme?
- ▶ Perfect secrecy: $E_K(m) \equiv E_K(m')$, for any $m, m' \in \{0, 1\}^\ell$ and $K \sim \{0, 1\}^n$, letting $E_k(x) := E(k, x)$.

Encryption schemes

Definition 1

A pair of algorithms (E, D) is (perfectly correct) encryption scheme, if for any $k \in \{0, 1\}^n$ and $m \in \{0, 1\}^\ell$, it holds that $D(k, E(k, m)) = m$

- ▶ What security should we ask from such scheme?
- ▶ Perfect secrecy: $E_K(m) \equiv E_K(m')$, for any $m, m' \in \{0, 1\}^\ell$ and $K \sim \{0, 1\}^n$, letting $E_k(x) := E(k, x)$.
- ▶ Theorem (Shannon): Perfect secrecy implies $n \geq \ell$.

Encryption schemes

Definition 1

A pair of algorithms (E, D) is (perfectly correct) encryption scheme, if for any $k \in \{0, 1\}^n$ and $m \in \{0, 1\}^\ell$, it holds that $D(k, E(k, m)) = m$

- ▶ What security should we ask from such scheme?
- ▶ Perfect secrecy: $E_K(m) \equiv E_K(m')$, for any $m, m' \in \{0, 1\}^\ell$ and $K \sim \{0, 1\}^n$, letting $E_k(x) := E(k, x)$.
- ▶ Theorem (Shannon): Perfect secrecy implies $n \geq \ell$.
- ▶ Is it bad?

Encryption schemes

Definition 1

A pair of algorithms (E, D) is (perfectly correct) encryption scheme, if for any $k \in \{0, 1\}^n$ and $m \in \{0, 1\}^\ell$, it holds that $D(k, E(k, m)) = m$

- ▶ What security should we ask from such scheme?
- ▶ Perfect secrecy: $E_K(m) \equiv E_K(m')$, for any $m, m' \in \{0, 1\}^\ell$ and $K \sim \{0, 1\}^n$, letting $E_k(x) := E(k, x)$.
- ▶ Theorem (Shannon): Perfect secrecy implies $n \geq \ell$.
- ▶ Is it bad?

Encryption schemes

Definition 1

A pair of algorithms (E, D) is (perfectly correct) encryption scheme, if for any $k \in \{0, 1\}^n$ and $m \in \{0, 1\}^\ell$, it holds that $D(k, E(k, m)) = m$

- ▶ What security should we ask from such scheme?
- ▶ Perfect secrecy: $E_K(m) \equiv E_K(m')$, for any $m, m' \in \{0, 1\}^\ell$ and $K \sim \{0, 1\}^n$, letting $E_k(x) := E(k, x)$.
- ▶ Theorem (Shannon): Perfect secrecy implies $n \geq \ell$.
- ▶ Is it bad? Is it optimal?

Encryption schemes

Definition 1

A pair of algorithms (E, D) is (perfectly correct) encryption scheme, if for any $k \in \{0, 1\}^n$ and $m \in \{0, 1\}^\ell$, it holds that $D(k, E(k, m)) = m$

- ▶ What security should we ask from such scheme?
- ▶ Perfect secrecy: $E_K(m) \equiv E_K(m')$, for any $m, m' \in \{0, 1\}^\ell$ and $K \sim \{0, 1\}^n$, letting $E_k(x) := E(k, x)$.
- ▶ Theorem (Shannon): Perfect secrecy implies $n \geq \ell$.
- ▶ Is it bad? Is it optimal?
- ▶ Proof:

Encryption schemes

Definition 1

A pair of algorithms (E, D) is (perfectly correct) encryption scheme, if for any $k \in \{0, 1\}^n$ and $m \in \{0, 1\}^\ell$, it holds that $D(k, E(k, m)) = m$

- ▶ What security should we ask from such scheme?
- ▶ Perfect secrecy: $E_K(m) \equiv E_K(m')$, for any $m, m' \in \{0, 1\}^\ell$ and $K \sim \{0, 1\}^n$, letting $E_k(x) := E(k, x)$.
- ▶ Theorem (Shannon): Perfect secrecy implies $n \geq \ell$.
- ▶ Is it bad? Is it optimal?
- ▶ Proof:

Encryption schemes

Definition 1

A pair of algorithms (E, D) is (perfectly correct) encryption scheme, if for any $k \in \{0, 1\}^n$ and $m \in \{0, 1\}^\ell$, it holds that $D(k, E(k, m)) = m$

- ▶ What security should we ask from such scheme?
- ▶ Perfect secrecy: $E_K(m) \equiv E_K(m')$, for any $m, m' \in \{0, 1\}^\ell$ and $K \sim \{0, 1\}^n$, letting $E_k(x) := E(k, x)$.
- ▶ Theorem (Shannon): Perfect secrecy implies $n \geq \ell$.
- ▶ Is it bad? Is it optimal?
- ▶ Proof: Let $M \sim \{0, 1\}^\ell$.

Encryption schemes

Definition 1

A pair of algorithms (E, D) is (perfectly correct) encryption scheme, if for any $k \in \{0, 1\}^n$ and $m \in \{0, 1\}^\ell$, it holds that $D(k, E(k, m)) = m$

- ▶ What security should we ask from such scheme?
- ▶ Perfect secrecy: $E_K(m) \equiv E_K(m')$, for any $m, m' \in \{0, 1\}^\ell$ and $K \sim \{0, 1\}^n$, letting $E_k(x) := E(k, x)$.
- ▶ Theorem (Shannon): Perfect secrecy implies $n \geq \ell$.
- ▶ Is it bad? Is it optimal?
- ▶ Proof: Let $M \sim \{0, 1\}^\ell$.
- ▶ Perfect secrecy $\implies H(M, E_K(M)) = H(M, E_K(0^\ell))$

Encryption schemes

Definition 1

A pair of algorithms (E, D) is (perfectly correct) encryption scheme, if for any $k \in \{0, 1\}^n$ and $m \in \{0, 1\}^\ell$, it holds that $D(k, E(k, m)) = m$

- ▶ What security should we ask from such scheme?
- ▶ Perfect secrecy: $E_K(m) \equiv E_K(m')$, for any $m, m' \in \{0, 1\}^\ell$ and $K \sim \{0, 1\}^n$, letting $E_k(x) := E(k, x)$.
- ▶ Theorem (Shannon): Perfect secrecy implies $n \geq \ell$.
- ▶ Is it bad? Is it optimal?
- ▶ Proof: Let $M \sim \{0, 1\}^\ell$.
- ▶ Perfect secrecy $\implies H(M, E_K(M)) = H(M, E_K(0^\ell))$
 $\implies H(M|E_K(M)) = H(M, E_K(M)) - H(E_K(M)) = H(M|E_K(0^\ell)) = \ell$

Encryption schemes

Definition 1

A pair of algorithms (E, D) is (perfectly correct) encryption scheme, if for any $k \in \{0, 1\}^n$ and $m \in \{0, 1\}^\ell$, it holds that $D(k, E(k, m)) = m$

- ▶ What security should we ask from such scheme?
- ▶ Perfect secrecy: $E_K(m) \equiv E_K(m')$, for any $m, m' \in \{0, 1\}^\ell$ and $K \sim \{0, 1\}^n$, letting $E_k(x) := E(k, x)$.
- ▶ Theorem (Shannon): Perfect secrecy implies $n \geq \ell$.
- ▶ Is it bad? Is it optimal?
- ▶ Proof: Let $M \sim \{0, 1\}^\ell$.
- ▶ Perfect secrecy $\implies H(M, E_K(M)) = H(M, E_K(0^\ell))$
 $\implies H(M|E_K(M)) = H(M, E_K(M)) - H(E_K(M)) = H(M|E_K(0^\ell)) = \ell$
- ▶ Perfect correctness $\implies H(M|E_K(M), K) = 0$

Encryption schemes

Definition 1

A pair of algorithms (E, D) is (perfectly correct) encryption scheme, if for any $k \in \{0, 1\}^n$ and $m \in \{0, 1\}^\ell$, it holds that $D(k, E(k, m)) = m$

- ▶ What security should we ask from such scheme?
- ▶ Perfect secrecy: $E_K(m) \equiv E_K(m')$, for any $m, m' \in \{0, 1\}^\ell$ and $K \sim \{0, 1\}^n$, letting $E_k(x) := E(k, x)$.
- ▶ Theorem (Shannon): Perfect secrecy implies $n \geq \ell$.
- ▶ Is it bad? Is it optimal?
- ▶ Proof: Let $M \sim \{0, 1\}^\ell$.
- ▶ Perfect secrecy $\implies H(M, E_K(M)) = H(M, E_K(0^\ell))$
 $\implies H(M|E_K(M)) = H(M, E_K(M)) - H(E_K(M)) = H(M|E_K(0^\ell)) = \ell$
- ▶ Perfect correctness $\implies H(M|E_K(M), K) = 0$
 $\implies H(M|E_K(M)) \leq H(M, K|E_K(M)) \leq H(K|E_K(M)) + 0 \leq H(K) = n$

Encryption schemes

Definition 1

A pair of algorithms (E, D) is (perfectly correct) encryption scheme, if for any $k \in \{0, 1\}^n$ and $m \in \{0, 1\}^\ell$, it holds that $D(k, E(k, m)) = m$

- ▶ What security should we ask from such scheme?
- ▶ Perfect secrecy: $E_K(m) \equiv E_K(m')$, for any $m, m' \in \{0, 1\}^\ell$ and $K \sim \{0, 1\}^n$, letting $E_k(x) := E(k, x)$.
- ▶ Theorem (Shannon): Perfect secrecy implies $n \geq \ell$.
- ▶ Is it bad? Is it optimal?
- ▶ Proof: Let $M \sim \{0, 1\}^\ell$.
- ▶ Perfect secrecy $\implies H(M, E_K(M)) = H(M, E_K(0^\ell))$
 $\implies H(M|E_K(M)) = H(M, E_K(M)) - H(E_K(M)) = H(M|E_K(0^\ell)) = \ell$
- ▶ Perfect correctness $\implies H(M|E_K(M), K) = 0$
 $\implies H(M|E_K(M)) \leq H(M, K|E_K(M)) \leq H(K|E_K(M)) + 0 \leq H(K) = n$
 $\implies \ell \leq n. \square$

Encryption schemes

Definition 1

A pair of algorithms (E, D) is (perfectly correct) encryption scheme, if for any $k \in \{0, 1\}^n$ and $m \in \{0, 1\}^\ell$, it holds that $D(k, E(k, m)) = m$

- ▶ What security should we ask from such scheme?
- ▶ Perfect secrecy: $E_K(m) \equiv E_K(m')$, for any $m, m' \in \{0, 1\}^\ell$ and $K \sim \{0, 1\}^n$, letting $E_k(x) := E(k, x)$.
- ▶ Theorem (Shannon): Perfect secrecy implies $n \geq \ell$.
- ▶ Is it bad? Is it optimal?
- ▶ Proof: Let $M \sim \{0, 1\}^\ell$.
- ▶ Perfect secrecy $\implies H(M, E_K(M)) = H(M, E_K(0^\ell))$
 $\implies H(M|E_K(M)) = H(M, E_K(M)) - H(E_K(M)) = H(M|E_K(0^\ell)) = \ell$
- ▶ Perfect correctness $\implies H(M|E_K(M), K) = 0$
 $\implies H(M|E_K(M)) \leq H(M, K|E_K(M)) \leq H(K|E_K(M)) + 0 \leq H(K) = n$
 $\implies \ell \leq n. \square$
- ▶ Statistical secrecy?

Encryption schemes

Definition 1

A pair of algorithms (E, D) is (perfectly correct) encryption scheme, if for any $k \in \{0, 1\}^n$ and $m \in \{0, 1\}^\ell$, it holds that $D(k, E(k, m)) = m$

- ▶ What security should we ask from such scheme?
- ▶ Perfect secrecy: $E_K(m) \equiv E_K(m')$, for any $m, m' \in \{0, 1\}^\ell$ and $K \sim \{0, 1\}^n$, letting $E_k(x) := E(k, x)$.
- ▶ Theorem (Shannon): Perfect secrecy implies $n \geq \ell$.
- ▶ Is it bad? Is it optimal?
- ▶ Proof: Let $M \sim \{0, 1\}^\ell$.
- ▶ Perfect secrecy $\implies H(M, E_K(M)) = H(M, E_K(0^\ell))$
 $\implies H(M|E_K(M)) = H(M, E_K(M)) - H(E_K(M)) = H(M|E_K(0^\ell)) = \ell$
- ▶ Perfect correctness $\implies H(M|E_K(M), K) = 0$
 $\implies H(M|E_K(M)) \leq H(M, K|E_K(M)) \leq H(K|E_K(M)) + 0 \leq H(K) = n$
 $\implies \ell \leq n. \square$
- ▶ Statistical secrecy?

Encryption schemes

Definition 1

A pair of algorithms (E, D) is (perfectly correct) encryption scheme, if for any $k \in \{0, 1\}^n$ and $m \in \{0, 1\}^\ell$, it holds that $D(k, E(k, m)) = m$

- ▶ What security should we ask from such scheme?
- ▶ Perfect secrecy: $E_K(m) \equiv E_K(m')$, for any $m, m' \in \{0, 1\}^\ell$ and $K \sim \{0, 1\}^n$, letting $E_k(x) := E(k, x)$.
- ▶ Theorem (Shannon): Perfect secrecy implies $n \geq \ell$.
- ▶ Is it bad? Is it optimal?
- ▶ Proof: Let $M \sim \{0, 1\}^\ell$.
- ▶ Perfect secrecy $\implies H(M, E_K(M)) = H(M, E_K(0^\ell))$
 $\implies H(M|E_K(M)) = H(M, E_K(M)) - H(E_K(M)) = H(M|E_K(0^\ell)) = \ell$
- ▶ Perfect correctness $\implies H(M|E_K(M), K) = 0$
 $\implies H(M|E_K(M)) \leq H(M, K|E_K(M)) \leq H(K|E_K(M)) + 0 \leq H(K) = n$
 $\implies \ell \leq n. \square$
- ▶ Statistical secrecy? HW.

Encryption schemes

Definition 1

A pair of algorithms (E, D) is (perfectly correct) encryption scheme, if for any $k \in \{0, 1\}^n$ and $m \in \{0, 1\}^\ell$, it holds that $D(k, E(k, m)) = m$

- ▶ What security should we ask from such scheme?
- ▶ Perfect secrecy: $E_K(m) \equiv E_K(m')$, for any $m, m' \in \{0, 1\}^\ell$ and $K \sim \{0, 1\}^n$, letting $E_k(x) := E(k, x)$.
- ▶ Theorem (Shannon): Perfect secrecy implies $n \geq \ell$.
- ▶ Is it bad? Is it optimal?
- ▶ Proof: Let $M \sim \{0, 1\}^\ell$.
- ▶ Perfect secrecy $\implies H(M, E_K(M)) = H(M, E_K(0^\ell))$
 $\implies H(M|E_K(M)) = H(M, E_K(M)) - H(E_K(M)) = H(M|E_K(0^\ell)) = \ell$
- ▶ Perfect correctness $\implies H(M|E_K(M), K) = 0$
 $\implies H(M|E_K(M)) \leq H(M, K|E_K(M)) \leq H(K|E_K(M)) + 0 \leq H(K) = n$
 $\implies \ell \leq n. \square$
- ▶ Statistical secrecy? HW. Computational secrecy?

Part II

Statistical Vs. Computational distance

Distributions and statistical distance

Let \mathcal{P} and \mathcal{Q} be two distributions over a finite set \mathcal{U} . Their **statistical distance** (also known as, variation distance) is defined as

$$\text{SD}(\mathcal{P}, \mathcal{Q}) := \frac{1}{2} \sum_{x \in \mathcal{U}} |\mathcal{P}(x) - \mathcal{Q}(x)| = \max_{\mathcal{S} \subseteq \mathcal{U}} (\mathcal{P}(\mathcal{S}) - \mathcal{Q}(\mathcal{S}))$$

We will only consider **finite** distributions.

Distributions and statistical distance

Let \mathcal{P} and \mathcal{Q} be two distributions over a finite set \mathcal{U} . Their **statistical distance** (also known as, variation distance) is defined as

$$\text{SD}(\mathcal{P}, \mathcal{Q}) := \frac{1}{2} \sum_{x \in \mathcal{U}} |\mathcal{P}(x) - \mathcal{Q}(x)| = \max_{S \subseteq \mathcal{U}} (\mathcal{P}(S) - \mathcal{Q}(S))$$

We will only consider **finite** distributions.

Claim 2

For any pair of (finite) distributions \mathcal{P} and \mathcal{Q} , it holds that

$$\text{SD}(\mathcal{P}, \mathcal{Q}) = \max_D \{ \Delta^D(\mathcal{P}, \mathcal{Q}) := \Pr_{x \leftarrow \mathcal{P}} [D(x) = 1] - \Pr_{x \leftarrow \mathcal{Q}} [D(x) = 1] \},$$

where D is **any** algorithm.

Distributions and statistical distance

Let \mathcal{P} and \mathcal{Q} be two distributions over a finite set \mathcal{U} . Their **statistical distance** (also known as, variation distance) is defined as

$$\text{SD}(\mathcal{P}, \mathcal{Q}) := \frac{1}{2} \sum_{x \in \mathcal{U}} |\mathcal{P}(x) - \mathcal{Q}(x)| = \max_{S \subseteq \mathcal{U}} (\mathcal{P}(S) - \mathcal{Q}(S))$$

We will only consider **finite** distributions.

Claim 2

For any pair of (finite) distributions \mathcal{P} and \mathcal{Q} , it holds that

$$\text{SD}(\mathcal{P}, \mathcal{Q}) = \max_{\mathcal{D}} \{ \Delta^{\mathcal{D}}(\mathcal{P}, \mathcal{Q}) := \Pr_{x \leftarrow \mathcal{P}} [\mathcal{D}(x) = 1] - \Pr_{x \leftarrow \mathcal{Q}} [\mathcal{D}(x) = 1] \},$$

where \mathcal{D} is **any** algorithm.

Let $\mathcal{P}, \mathcal{Q}, \mathcal{R}$ be finite distributions, then

Triangle inequality: $\text{SD}(\mathcal{P}, \mathcal{R}) \leq \text{SD}(\mathcal{P}, \mathcal{Q}) + \text{SD}(\mathcal{Q}, \mathcal{R})$

Repeated sampling: $\text{SD}(\mathcal{P}^2 = (\mathcal{P}, \mathcal{P}), \mathcal{Q}^2 = (\mathcal{Q}, \mathcal{Q})) \leq 2 \cdot \text{SD}(\mathcal{P}, \mathcal{Q})$

Section 1

Computational Indistinguishability

Computational indistinguishability

Definition 3 (computational indistinguishability)

\mathcal{P} and \mathcal{Q} are (s, ϵ) -indistinguishable, if $\Delta_{\mathcal{P}, \mathcal{Q}}^D \leq \epsilon$, for any s -size D .

Computational indistinguishability

Definition 3 (computational indistinguishability)

\mathcal{P} and \mathcal{Q} are (s, ϵ) -indistinguishable, if $\Delta_{\mathcal{P}, \mathcal{Q}}^D \leq \epsilon$, for any s -size D .

- Adversaries are circuits (possibly randomized)

Computational indistinguishability

Definition 3 (computational indistinguishability)

\mathcal{P} and \mathcal{Q} are (s, ε) -indistinguishable, if $\Delta_{\mathcal{P}, \mathcal{Q}}^D \leq \varepsilon$, for any s -size D .

- ▶ Adversaries are circuits (possibly randomized)
- ▶ (∞, ε) -indistinguishable is equivalent to statistical distance ε

Computational indistinguishability

Definition 3 (computational indistinguishability)

\mathcal{P} and \mathcal{Q} are (s, ϵ) -indistinguishable, if $\Delta_{\mathcal{P}, \mathcal{Q}}^D \leq \epsilon$, for any s -size D .

- ▶ Adversaries are circuits (possibly randomized)
- ▶ (∞, ϵ) -indistinguishable is equivalent to statistical distance ϵ
- ▶ Can it be different from the statistical case?

Computational indistinguishability

Definition 3 (computational indistinguishability)

\mathcal{P} and \mathcal{Q} are (s, ε) -indistinguishable, if $\Delta_{\mathcal{P}, \mathcal{Q}}^D \leq \varepsilon$, for any s -size D .

- ▶ Adversaries are circuits (possibly randomized)
- ▶ (∞, ε) -indistinguishable is equivalent to statistical distance ε
- ▶ Can it be different from the statistical case?
- ▶ Unless said otherwise, distributions are over $\{0, 1\}^n$

Repeated sampling

Question 4

Assume \mathcal{P} and \mathcal{Q} are (s, ϵ) -indistinguishable, what about \mathcal{P}^2 and \mathcal{Q}^2 ?

Repeated sampling

Question 4

Assume \mathcal{P} and \mathcal{Q} are (s, ε) -indistinguishable, what about \mathcal{P}^2 and \mathcal{Q}^2 ?

- ▶ Let D be an s' -size algorithm with $\Delta^D(\mathcal{P}^2, \mathcal{Q}^2) = \varepsilon'$

Repeated sampling

Question 4

Assume \mathcal{P} and \mathcal{Q} are (s, ε) -indistinguishable, what about \mathcal{P}^2 and \mathcal{Q}^2 ?

- ▶ Let D be an s' -size algorithm with $\Delta^D(\mathcal{P}^2, \mathcal{Q}^2) = \varepsilon'$

Repeated sampling

Question 4

Assume \mathcal{P} and \mathcal{Q} are (s, ε) -indistinguishable, what about \mathcal{P}^2 and \mathcal{Q}^2 ?

- ▶ Let D be an s' -size algorithm with $\Delta^D(\mathcal{P}^2, \mathcal{Q}^2) = \varepsilon'$

$$\begin{aligned}\varepsilon' &= \Pr_{x \leftarrow \mathcal{P}^2} [D(x) = 1] - \Pr_{x \leftarrow \mathcal{Q}^2} [D(x) = 1] \\ &= \left(\Pr_{x \leftarrow \mathcal{P}^2} [D(x) = 1] - \Pr_{x \leftarrow (\mathcal{P}, \mathcal{Q})} [D(x) = 1] \right) \\ &\quad + \left(\Pr_{x \leftarrow (\mathcal{P}, \mathcal{Q})} [D(x) = 1] - \Pr_{x \leftarrow \mathcal{Q}^2} [D(x) = 1] \right)\end{aligned}$$

Repeated sampling

Question 4

Assume \mathcal{P} and \mathcal{Q} are (s, ε) -indistinguishable, what about \mathcal{P}^2 and \mathcal{Q}^2 ?

- ▶ Let D be an s' -size algorithm with $\Delta^D(\mathcal{P}^2, \mathcal{Q}^2) = \varepsilon'$

$$\begin{aligned}\varepsilon' &= \Pr_{x \leftarrow \mathcal{P}^2} [D(x) = 1] - \Pr_{x \leftarrow \mathcal{Q}^2} [D(x) = 1] \\ &= \left(\Pr_{x \leftarrow \mathcal{P}^2} [D(x) = 1] - \Pr_{x \leftarrow (\mathcal{P}, \mathcal{Q})} [D(x) = 1] \right) \\ &\quad + \left(\Pr_{x \leftarrow (\mathcal{P}, \mathcal{Q})} [D(x) = 1] - \Pr_{x \leftarrow \mathcal{Q}^2} [D(x) = 1] \right) \\ &= \Delta^D(\mathcal{P}^2, (\mathcal{P}, \mathcal{Q})) + \Delta^D((\mathcal{P}, \mathcal{Q}), \mathcal{Q}^2)\end{aligned}$$

Repeated sampling

Question 4

Assume \mathcal{P} and \mathcal{Q} are (s, ε) -indistinguishable, what about \mathcal{P}^2 and \mathcal{Q}^2 ?

- ▶ Let D be an s' -size algorithm with $\Delta^D(\mathcal{P}^2, \mathcal{Q}^2) = \varepsilon'$

$$\begin{aligned}\varepsilon' &= \Pr_{x \leftarrow \mathcal{P}^2} [D(x) = 1] - \Pr_{x \leftarrow \mathcal{Q}^2} [D(x) = 1] \\ &= \left(\Pr_{x \leftarrow \mathcal{P}^2} [D(x) = 1] - \Pr_{x \leftarrow (\mathcal{P}, \mathcal{Q})} [D(x) = 1] \right) \\ &\quad + \left(\Pr_{x \leftarrow (\mathcal{P}, \mathcal{Q})} [D(x) = 1] - \Pr_{x \leftarrow \mathcal{Q}^2} [D(x) = 1] \right) \\ &= \Delta^D(\mathcal{P}^2, (\mathcal{P}, \mathcal{Q})) + \Delta^D((\mathcal{P}, \mathcal{Q}), \mathcal{Q}^2)\end{aligned}$$

- ▶ So either $\Delta^D(\mathcal{P}^2, (\mathcal{P}, \mathcal{Q})) \geq \varepsilon'/2$, or $\Delta^D((\mathcal{P}, \mathcal{Q}), \mathcal{Q}^2) \geq \varepsilon'/2$

Repeated sampling

Question 4

Assume \mathcal{P} and \mathcal{Q} are (s, ε) -indistinguishable, what about \mathcal{P}^2 and \mathcal{Q}^2 ?

- ▶ Let D be an s' -size algorithm with $\Delta^D(\mathcal{P}^2, \mathcal{Q}^2) = \varepsilon'$

$$\begin{aligned}\varepsilon' &= \Pr_{x \leftarrow \mathcal{P}^2} [D(x) = 1] - \Pr_{x \leftarrow \mathcal{Q}^2} [D(x) = 1] \\ &= \left(\Pr_{x \leftarrow \mathcal{P}^2} [D(x) = 1] - \Pr_{x \leftarrow (\mathcal{P}, \mathcal{Q})} [D(x) = 1] \right) \\ &\quad + \left(\Pr_{x \leftarrow (\mathcal{P}, \mathcal{Q})} [D(x) = 1] - \Pr_{x \leftarrow \mathcal{Q}^2} [D(x) = 1] \right) \\ &= \Delta^D(\mathcal{P}^2, (\mathcal{P}, \mathcal{Q})) + \Delta^D((\mathcal{P}, \mathcal{Q}), \mathcal{Q}^2)\end{aligned}$$

- ▶ So either $\Delta^D(\mathcal{P}^2, (\mathcal{P}, \mathcal{Q})) \geq \varepsilon'/2$, or $\Delta^D((\mathcal{P}, \mathcal{Q}), \mathcal{Q}^2) \geq \varepsilon'/2$
- ▶ Hence, $\varepsilon' < 2\varepsilon$ implies $s' \geq s - n$. (?)

Repeated sampling cont.

What about \mathcal{P}^k and \mathcal{Q}^k ?

Repeated sampling cont.

What about \mathcal{P}^k and \mathcal{Q}^k ?

Claim 5

Assume \mathcal{P} and \mathcal{Q} are (s, ϵ) -indistinguishable, then \mathcal{P}^k and \mathcal{Q}^k are $(s - kn, k\epsilon)$ -indistinguishable.

Repeated sampling cont.

What about \mathcal{P}^k and \mathcal{Q}^k ?

Claim 5

Assume \mathcal{P} and \mathcal{Q} are (s, ϵ) -indistinguishable, then \mathcal{P}^k and \mathcal{Q}^k are $(s - kn, k\epsilon)$ -indistinguishable.

Proof: ?

Repeated sampling cont.

What about \mathcal{P}^k and \mathcal{Q}^k ?

Claim 5

Assume \mathcal{P} and \mathcal{Q} are (s, ϵ) -indistinguishable, then \mathcal{P}^k and \mathcal{Q}^k are $(s - kn, k\epsilon)$ -indistinguishable.

Proof: ?

- ▶ For $i \in \{0, \dots, k\}$, let $H^i = (P_1, \dots, P_i, Q_{i+1}, \dots, Q_k)$, where the P_i 's are iid $\sim \mathcal{P}$ and the Q_i 's are iid $\sim \mathcal{Q}$. (hybrids)

Repeated sampling cont.

What about \mathcal{P}^k and \mathcal{Q}^k ?

Claim 5

Assume \mathcal{P} and \mathcal{Q} are (s, ϵ) -indistinguishable, then \mathcal{P}^k and \mathcal{Q}^k are $(s - kn, k\epsilon)$ -indistinguishable.

Proof: ?

- ▶ For $i \in \{0, \dots, k\}$, let $H^i = (P_1, \dots, P_i, Q_{i+1}, \dots, Q_k)$, where the P_i 's are iid $\sim \mathcal{P}$ and the Q_i 's are iid $\sim \mathcal{Q}$. (hybrids)
- ▶ Let D be a s' -size algorithm with $\Delta^D(\mathcal{P}^k, \mathcal{Q}^k) = \epsilon'$

Repeated sampling cont.

What about \mathcal{P}^k and \mathcal{Q}^k ?

Claim 5

Assume \mathcal{P} and \mathcal{Q} are (s, ϵ) -indistinguishable, then \mathcal{P}^k and \mathcal{Q}^k are $(s - kn, k\epsilon)$ -indistinguishable.

Proof: ?

- ▶ For $i \in \{0, \dots, k\}$, let $H^i = (P_1, \dots, P_i, Q_{i+1}, \dots, Q_k)$, where the P_i 's are iid $\sim \mathcal{P}$ and the Q_i 's are iid $\sim \mathcal{Q}$. (hybrids)
- ▶ Let D be a s' -size algorithm with $\Delta^D(\mathcal{P}^k, \mathcal{Q}^k) = \epsilon'$
- ▶ $\epsilon' = \Pr [D(H^k) = 1] - \Pr [D(H^0) = 1]$.

Repeated sampling cont.

What about \mathcal{P}^k and \mathcal{Q}^k ?

Claim 5

Assume \mathcal{P} and \mathcal{Q} are (s, ε) -indistinguishable, then \mathcal{P}^k and \mathcal{Q}^k are $(s - kn, k\varepsilon)$ -indistinguishable.

Proof: ?

- ▶ For $i \in \{0, \dots, k\}$, let $H^i = (P_1, \dots, P_i, Q_{i+1}, \dots, Q_k)$, where the P_i 's are iid $\sim \mathcal{P}$ and the Q_i 's are iid $\sim \mathcal{Q}$. (hybrids)
- ▶ Let D be a s' -size algorithm with $\Delta^D(\mathcal{P}^k, \mathcal{Q}^k) = \varepsilon'$
- ▶ $\varepsilon' = \Pr [D(H^k) = 1] - \Pr [D(H^0) = 1]$.
- ▶ $\varepsilon' = \sum_{i \in [k]} \Pr [D(H^i) = 1] - \Pr [D(H^{i-1}) = 1]$

Repeated sampling cont.

What about \mathcal{P}^k and \mathcal{Q}^k ?

Claim 5

Assume \mathcal{P} and \mathcal{Q} are (s, ε) -indistinguishable, then \mathcal{P}^k and \mathcal{Q}^k are $(s - kn, k\varepsilon)$ -indistinguishable.

Proof: ?

- ▶ For $i \in \{0, \dots, k\}$, let $H^i = (P_1, \dots, P_i, Q_{i+1}, \dots, Q_k)$, where the P_i 's are iid $\sim \mathcal{P}$ and the Q_i 's are iid $\sim \mathcal{Q}$. (hybrids)
- ▶ Let D be a s' -size algorithm with $\Delta^D(\mathcal{P}^k, \mathcal{Q}^k) = \varepsilon'$
- ▶ $\varepsilon' = \Pr [D(H^k) = 1] - \Pr [D(H^0) = 1]$.
- ▶ $\varepsilon' = \sum_{i \in [k]} \Pr [D(H^i) = 1] - \Pr [D(H^{i-1}) = 1]$

Repeated sampling cont.

What about \mathcal{P}^k and \mathcal{Q}^k ?

Claim 5

Assume \mathcal{P} and \mathcal{Q} are (s, ε) -indistinguishable, then \mathcal{P}^k and \mathcal{Q}^k are $(s - kn, k\varepsilon)$ -indistinguishable.

Proof: ?

- ▶ For $i \in \{0, \dots, k\}$, let $H^i = (P_1, \dots, P_i, Q_{i+1}, \dots, Q_k)$, where the P_i 's are iid $\sim \mathcal{P}$ and the Q_i 's are iid $\sim \mathcal{Q}$. (hybrids)
- ▶ Let D be a s' -size algorithm with $\Delta^D(\mathcal{P}^k, \mathcal{Q}^k) = \varepsilon'$
- ▶ $\varepsilon' = \Pr [D(H^k) = 1] - \Pr [D(H^0) = 1]$.
- ▶ $\varepsilon' = \sum_{i \in [k]} \Pr [D(H^i) = 1] - \Pr [D(H^{i-1}) = 1] = \sum_{i \in [k]} \Delta^D(H^i, H^{i-1})$

Repeated sampling cont.

What about \mathcal{P}^k and \mathcal{Q}^k ?

Claim 5

Assume \mathcal{P} and \mathcal{Q} are (s, ε) -indistinguishable, then \mathcal{P}^k and \mathcal{Q}^k are $(s - kn, k\varepsilon)$ -indistinguishable.

Proof: ?

- ▶ For $i \in \{0, \dots, k\}$, let $H^i = (P_1, \dots, P_i, Q_{i+1}, \dots, Q_k)$, where the P_i 's are iid $\sim \mathcal{P}$ and the Q_i 's are iid $\sim \mathcal{Q}$. (hybrids)
- ▶ Let D be a s' -size algorithm with $\Delta^D(\mathcal{P}^k, \mathcal{Q}^k) = \varepsilon'$
- ▶ $\varepsilon' = \Pr [D(H^k) = 1] - \Pr [D(H^0) = 1]$.
- ▶ $\varepsilon' = \sum_{i \in [k]} \Pr [D(H^i) = 1] - \Pr [D(H^{i-1}) = 1] = \sum_{i \in [k]} \Delta^D(H^i, H^{i-1})$
- ▶ $\implies \exists i \in [k]$ with $\Delta^D(H^i, H^{i-1}) \geq \varepsilon'/k$.

Repeated sampling cont.

What about \mathcal{P}^k and \mathcal{Q}^k ?

Claim 5

Assume \mathcal{P} and \mathcal{Q} are (s, ε) -indistinguishable, then \mathcal{P}^k and \mathcal{Q}^k are $(s - kn, k\varepsilon)$ -indistinguishable.

Proof: ?

- ▶ For $i \in \{0, \dots, k\}$, let $H^i = (P_1, \dots, P_i, Q_{i+1}, \dots, Q_k)$, where the P_i 's are iid $\sim \mathcal{P}$ and the Q_i 's are iid $\sim \mathcal{Q}$. (hybrids)
- ▶ Let D be a s' -size algorithm with $\Delta^D(\mathcal{P}^k, \mathcal{Q}^k) = \varepsilon'$
- ▶ $\varepsilon' = \Pr [D(H^k) = 1] - \Pr [D(H^0) = 1]$.
- ▶ $\varepsilon' = \sum_{i \in [k]} \Pr [D(H^i) = 1] - \Pr [D(H^{i-1}) = 1] = \sum_{i \in [k]} \Delta^D(H^i, H^{i-1})$
- ▶ $\implies \exists i \in [k]$ with $\Delta^D(H^i, H^{i-1}) \geq \varepsilon'/k$.
- ▶ Thus, $\varepsilon' \leq k\varepsilon$ implies $s' > s - kn$

Repeated sampling cont.

What about \mathcal{P}^k and \mathcal{Q}^k ?

Claim 5

Assume \mathcal{P} and \mathcal{Q} are (s, ε) -indistinguishable, then \mathcal{P}^k and \mathcal{Q}^k are $(s - kn, k\varepsilon)$ -indistinguishable.

Proof: ?

- ▶ For $i \in \{0, \dots, k\}$, let $H^i = (P_1, \dots, P_i, Q_{i+1}, \dots, Q_k)$, where the P_i 's are iid $\sim \mathcal{P}$ and the Q_i 's are iid $\sim \mathcal{Q}$. (hybrids)
- ▶ Let D be a s' -size algorithm with $\Delta^D(\mathcal{P}^k, \mathcal{Q}^k) = \varepsilon'$
- ▶ $\varepsilon' = \Pr [D(H^k) = 1] - \Pr [D(H^0) = 1]$.
- ▶ $\varepsilon' = \sum_{i \in [k]} \Pr [D(H^i) = 1] - \Pr [D(H^{i-1}) = 1] = \sum_{i \in [k]} \Delta^D(H^i, H^{i-1})$
- ▶ $\implies \exists i \in [k]$ with $\Delta^D(H^i, H^{i-1}) \geq \varepsilon'/k$.
- ▶ Thus, $\varepsilon' \leq k\varepsilon$ implies $s' > s - kn$
- ▶ When considering bounded **time** algorithms, things behaves very differently!

Part III

Pseudorandom Generators

Pseudorandom generator

Definition 6 (pseudorandom distributions)

A distribution \mathcal{P} over $\{0, 1\}^n$ is (s, ϵ) -pseudorandom, if it is (s, ϵ) -indistinguishable from U_n .

Pseudorandom generator

Definition 6 (pseudorandom distributions)

A distribution \mathcal{P} over $\{0, 1\}^n$ is (s, ϵ) -pseudorandom, if it is (s, ϵ) -indistinguishable from U_n .

- Do such distributions exist for interesting (s, ϵ)

Pseudorandom generator

Definition 6 (pseudorandom distributions)

A distribution \mathcal{P} over $\{0, 1\}^n$ is (s, ϵ) -pseudorandom, if it is (s, ϵ) -indistinguishable from U_n .

- Do such distributions exist for interesting (s, ϵ)

Pseudorandom generator

Definition 6 (pseudorandom distributions)

A distribution \mathcal{P} over $\{0, 1\}^n$ is (s, ε) -pseudorandom, if it is (s, ε) -indistinguishable from U_n .

- Do such distributions exist for interesting (s, ε)

Definition 7 (pseudorandom generators (PRGs))

A poly-time computable function $g: \{0, 1\}^n \mapsto \{0, 1\}^{\ell(n)}$ is a (s, ε) -pseudorandom generator, if for any $n \in \mathbb{N}$

Pseudorandom generator

Definition 6 (pseudorandom distributions)

A distribution \mathcal{P} over $\{0, 1\}^n$ is (s, ϵ) -pseudorandom, if it is (s, ϵ) -indistinguishable from U_n .

- ▶ Do such distributions exist for interesting (s, ϵ)

Definition 7 (pseudorandom generators (PRGs))

A poly-time computable function $g: \{0, 1\}^n \mapsto \{0, 1\}^{\ell(n)}$ is a (s, ϵ) -pseudorandom generator, if for any $n \in \mathbb{N}$

- ▶ g is length extending (i.e., $\ell(n) > n$)

Pseudorandom generator

Definition 6 (pseudorandom distributions)

A distribution \mathcal{P} over $\{0, 1\}^n$ is (s, ε) -pseudorandom, if it is (s, ε) -indistinguishable from U_n .

- ▶ Do such distributions exist for interesting (s, ε)

Definition 7 (pseudorandom generators (PRGs))

A poly-time computable function $g: \{0, 1\}^n \mapsto \{0, 1\}^{\ell(n)}$ is a (s, ε) -pseudorandom generator, if for any $n \in \mathbb{N}$

- ▶ g is length extending (i.e., $\ell(n) > n$)
- ▶ $g(U_n)$ is $(s(n), \varepsilon(n))$ -pseudorandom

Pseudorandom generator

Definition 6 (pseudorandom distributions)

A distribution \mathcal{P} over $\{0, 1\}^n$ is (s, ε) -pseudorandom, if it is (s, ε) -indistinguishable from U_n .

- ▶ Do such distributions exist for interesting (s, ε)

Definition 7 (pseudorandom generators (PRGs))

A poly-time computable function $g: \{0, 1\}^n \mapsto \{0, 1\}^{\ell(n)}$ is a (s, ε) -pseudorandom generator, if for any $n \in \mathbb{N}$

- ▶ g is length extending (i.e., $\ell(n) > n$)
- ▶ $g(U_n)$ is $(s(n), \varepsilon(n))$ -pseudorandom

Pseudorandom generator

Definition 6 (pseudorandom distributions)

A distribution \mathcal{P} over $\{0, 1\}^n$ is (s, ε) -pseudorandom, if it is (s, ε) -indistinguishable from U_n .

- ▶ Do such distributions exist for interesting (s, ε)

Definition 7 (pseudorandom generators (PRGs))

A poly-time computable function $g: \{0, 1\}^n \mapsto \{0, 1\}^{\ell(n)}$ is a (s, ε) -pseudorandom generator, if for any $n \in \mathbb{N}$

- ▶ g is length extending (i.e., $\ell(n) > n$)
- ▶ $g(U_n)$ is $(s(n), \varepsilon(n))$ -pseudorandom
- ▶ We omit the “security parameter”, i.e., n , when its value is clear from the context

Pseudorandom generator

Definition 6 (pseudorandom distributions)

A distribution \mathcal{P} over $\{0, 1\}^n$ is (s, ε) -pseudorandom, if it is (s, ε) -indistinguishable from U_n .

- ▶ Do such distributions exist for interesting (s, ε)

Definition 7 (pseudorandom generators (PRGs))

A poly-time computable function $g: \{0, 1\}^n \mapsto \{0, 1\}^{\ell(n)}$ is a (s, ε) -pseudorandom generator, if for any $n \in \mathbb{N}$

- ▶ g is length extending (i.e., $\ell(n) > n$)
- ▶ $g(U_n)$ is $(s(n), \varepsilon(n))$ -pseudorandom
- ▶ We omit the “security parameter”, i.e., n , when its value is clear from the context
- ▶ Do such generators exist?

Pseudorandom generator

Definition 6 (pseudorandom distributions)

A distribution \mathcal{P} over $\{0, 1\}^n$ is (s, ε) -pseudorandom, if it is (s, ε) -indistinguishable from U_n .

- ▶ Do such distributions exist for interesting (s, ε)

Definition 7 (pseudorandom generators (PRGs))

A poly-time computable function $g: \{0, 1\}^n \mapsto \{0, 1\}^{\ell(n)}$ is a (s, ε) -pseudorandom generator, if for any $n \in \mathbb{N}$

- ▶ g is length extending (i.e., $\ell(n) > n$)
- ▶ $g(U_n)$ is $(s(n), \varepsilon(n))$ -pseudorandom
- ▶ We omit the “security parameter”, i.e., n , when its value is clear from the context
- ▶ Do such generators exist?
- ▶ Applications?

Section 2

Pseudorandom generators (PRGs) from One-Way Permutations (OWPs)

OWP to PRG

Claim 8

Let $f: \{0, 1\}^n \mapsto \{0, 1\}^n$ be a poly-time permutation and let $b: \{0, 1\}^n \mapsto \{0, 1\}$ be a poly-time (s, ε) -hardcore predicate of f , then $g(x) = (f(x), b(x))$ is a $(s - O(n), \varepsilon)$ -PRG.

OWP to PRG

Claim 8

Let $f: \{0, 1\}^n \mapsto \{0, 1\}^n$ be a poly-time permutation and let $b: \{0, 1\}^n \mapsto \{0, 1\}$ be a poly-time (s, ε) -hardcore predicate of f , then $g(x) = (f(x), b(x))$ is a $(s - O(n), \varepsilon)$ -PRG.

► Hence, OWP \implies PRG

OWP to PRG

Claim 8

Let $f: \{0, 1\}^n \mapsto \{0, 1\}^n$ be a poly-time permutation and let $b: \{0, 1\}^n \mapsto \{0, 1\}$ be a poly-time (s, ε) -hardcore predicate of f , then $g(x) = (f(x), b(x))$ is a $(s - O(n), \varepsilon)$ -PRG.

- ▶ Hence, OWP \implies PRG
- ▶ Proof: Let D be an s' -size algorithm with $\Delta^D(g(U_n), U_{n+1}) = \varepsilon'$, we will show $\exists (s' + O(n))$ -size P with $\Pr[P(f(U_n)) = b(U_n)] = \frac{1}{2} + \varepsilon'$.

OWP to PRG

Claim 8

Let $f: \{0, 1\}^n \mapsto \{0, 1\}^n$ be a poly-time permutation and let $b: \{0, 1\}^n \mapsto \{0, 1\}$ be a poly-time (s, ε) -hardcore predicate of f , then $g(x) = (f(x), b(x))$ is a $(s - O(n), \varepsilon)$ -PRG.

- ▶ Hence, OWP \implies PRG
- ▶ Proof: Let D be an s' -size algorithm with $\Delta^D(g(U_n), U_{n+1}) = \varepsilon'$, we will show $\exists (s' + O(n))$ -size P with $\Pr[P(f(U_n)) = b(U_n)] = \frac{1}{2} + \varepsilon'$.
- ▶ Let $\delta = \Pr[D(U_{n+1}) = 1]$ (hence, $\Pr[D(g(U_n)) = 1] = \delta + \varepsilon'$)

OWP to PRG

Claim 8

Let $f: \{0, 1\}^n \mapsto \{0, 1\}^n$ be a poly-time permutation and let $b: \{0, 1\}^n \mapsto \{0, 1\}$ be a poly-time (s, ε) -hardcore predicate of f , then $g(x) = (f(x), b(x))$ is a $(s - O(n), \varepsilon)$ -PRG.

- ▶ Hence, OWP \implies PRG
- ▶ Proof: Let D be an s' -size algorithm with $\Delta^D(g(U_n), U_{n+1}) = \varepsilon'$, we will show $\exists (s' + O(n))$ -size P with $\Pr[P(f(U_n)) = b(U_n)] = \frac{1}{2} + \varepsilon'$.
- ▶ Let $\delta = \Pr[D(U_{n+1}) = 1]$ (hence, $\Pr[D(g(U_n)) = 1] = \delta + \varepsilon'$)
- ▶ Compute

$$\delta = \Pr[D(f(U_n), U_1) = 1]$$

OWP to PRG

Claim 8

Let $f: \{0, 1\}^n \mapsto \{0, 1\}^n$ be a poly-time permutation and let $b: \{0, 1\}^n \mapsto \{0, 1\}$ be a poly-time (s, ε) -hardcore predicate of f , then $g(x) = (f(x), b(x))$ is a $(s - O(n), \varepsilon)$ -PRG.

- ▶ Hence, OWP \implies PRG
- ▶ Proof: Let D be an s' -size algorithm with $\Delta^D(g(U_n), U_{n+1}) = \varepsilon'$, we will show $\exists (s' + O(n))$ -size P with $\Pr[P(f(U_n)) = b(U_n)] = \frac{1}{2} + \varepsilon'$.
- ▶ Let $\delta = \Pr[D(U_{n+1}) = 1]$ (hence, $\Pr[D(g(U_n)) = 1] = \delta + \varepsilon'$)
- ▶ Compute

$$\delta = \Pr[D(f(U_n), U_1) = 1]$$

OWP to PRG

Claim 8

Let $f: \{0, 1\}^n \mapsto \{0, 1\}^n$ be a poly-time permutation and let $b: \{0, 1\}^n \mapsto \{0, 1\}$ be a poly-time (s, ε) -hardcore predicate of f , then $g(x) = (f(x), b(x))$ is a $(s - O(n), \varepsilon)$ -PRG.

- ▶ Hence, OWP \implies PRG
- ▶ Proof: Let D be an s' -size algorithm with $\Delta^D(g(U_n), U_{n+1}) = \varepsilon'$, we will show $\exists (s' + O(n))$ -size P with $\Pr[P(f(U_n)) = b(U_n)] = \frac{1}{2} + \varepsilon'$.
- ▶ Let $\delta = \Pr[D(U_{n+1}) = 1]$ (hence, $\Pr[D(g(U_n)) = 1] = \delta + \varepsilon'$)
- ▶ Compute

$$\delta = \Pr[D(f(U_n), U_1) = 1] \quad (f \text{ is a permutation})$$

OWP to PRG

Claim 8

Let $f: \{0, 1\}^n \mapsto \{0, 1\}^n$ be a poly-time permutation and let $b: \{0, 1\}^n \mapsto \{0, 1\}$ be a poly-time (s, ε) -hardcore predicate of f , then $g(x) = (f(x), b(x))$ is a $(s - O(n), \varepsilon)$ -PRG.

- ▶ Hence, OWP \implies PRG
- ▶ Proof: Let D be an s' -size algorithm with $\Delta^D(g(U_n), U_{n+1}) = \varepsilon'$, we will show $\exists (s' + O(n))$ -size P with $\Pr[P(f(U_n)) = b(U_n)] = \frac{1}{2} + \varepsilon'$.
- ▶ Let $\delta = \Pr[D(U_{n+1}) = 1]$ (hence, $\Pr[D(g(U_n)) = 1] = \delta + \varepsilon'$)
- ▶ Compute

$$\begin{aligned}\delta &= \Pr[D(f(U_n), U_1) = 1] \quad (f \text{ is a permutation}) \\ &= \Pr[U_1 = b(U_n)] \cdot \Pr[D(f(U_n), U_1) = 1 \mid U_1 = b(U_n)] \\ &\quad + \Pr[U_1 = \overline{b(U_n)}] \cdot \Pr[D(f(U_n), U_1) = 1 \mid U_1 = \overline{b(U_n)}]\end{aligned}$$

OWP to PRG

Claim 8

Let $f: \{0, 1\}^n \mapsto \{0, 1\}^n$ be a poly-time permutation and let $b: \{0, 1\}^n \mapsto \{0, 1\}$ be a poly-time (s, ε) -hardcore predicate of f , then $g(x) = (f(x), b(x))$ is a $(s - O(n), \varepsilon)$ -PRG.

- ▶ Hence, OWP \implies PRG
- ▶ Proof: Let D be an s' -size algorithm with $\Delta^D(g(U_n), U_{n+1}) = \varepsilon'$, we will show $\exists (s' + O(n))$ -size P with $\Pr[P(f(U_n)) = b(U_n)] = \frac{1}{2} + \varepsilon'$.
- ▶ Let $\delta = \Pr[D(U_{n+1}) = 1]$ (hence, $\Pr[D(g(U_n)) = 1] = \delta + \varepsilon'$)
- ▶ Compute

$$\begin{aligned}\delta &= \Pr[D(f(U_n), U_1) = 1] \quad (f \text{ is a permutation}) \\ &= \Pr[U_1 = b(U_n)] \cdot \Pr[D(f(U_n), U_1) = 1 \mid U_1 = b(U_n)] \\ &\quad + \Pr[U_1 = \overline{b(U_n)}] \cdot \Pr[D(f(U_n), U_1) = 1 \mid U_1 = \overline{b(U_n)}] \\ &= \frac{1}{2}(\delta + \varepsilon') + \frac{1}{2} \cdot \Pr[D(f(U_n), U_1) = 1 \mid U_1 = \overline{b(U_n)}].\end{aligned}$$

OWP to PRG

Claim 8

Let $f: \{0, 1\}^n \mapsto \{0, 1\}^n$ be a poly-time permutation and let $b: \{0, 1\}^n \mapsto \{0, 1\}$ be a poly-time (s, ε) -hardcore predicate of f , then $g(x) = (f(x), b(x))$ is a $(s - O(n), \varepsilon)$ -PRG.

- ▶ Hence, OWP \implies PRG
- ▶ Proof: Let D be an s' -size algorithm with $\Delta^D(g(U_n), U_{n+1}) = \varepsilon'$, we will show $\exists (s' + O(n))$ -size P with $\Pr[P(f(U_n)) = b(U_n)] = \frac{1}{2} + \varepsilon'$.
- ▶ Let $\delta = \Pr[D(U_{n+1}) = 1]$ (hence, $\Pr[D(g(U_n)) = 1] = \delta + \varepsilon'$)
- ▶ Compute

$$\begin{aligned}\delta &= \Pr[D(f(U_n), U_1) = 1] \quad (f \text{ is a permutation}) \\ &= \Pr[U_1 = b(U_n)] \cdot \Pr[D(f(U_n), U_1) = 1 \mid U_1 = b(U_n)] \\ &\quad + \Pr[U_1 = \overline{b(U_n)}] \cdot \Pr[D(f(U_n), U_1) = 1 \mid U_1 = \overline{b(U_n)}] \\ &= \frac{1}{2}(\delta + \varepsilon') + \frac{1}{2} \cdot \Pr[D(f(U_n), U_1) = 1 \mid U_1 = \overline{b(U_n)}].\end{aligned}$$

- ▶ Hence, $\Pr[D(f(U_n), \overline{b(U_n)}) = 1] = \delta - \varepsilon'$

OWP to PRG cont.

- ▶ $\Pr[D(f(U_n), b(U_n)) = 1] = \delta + \varepsilon'$
- ▶ $\Pr[D(f(U_n), \overline{b(U_n)}) = 1] = \delta - \varepsilon'$

OWP to PRG cont.

- ▶ $\Pr[D(f(U_n), b(U_n)) = 1] = \delta + \varepsilon'$
- ▶ $\Pr[D(f(U_n), \overline{b(U_n)}) = 1] = \delta - \varepsilon'$

Algorithm 9 (P)

Input: $y \in \{0, 1\}^n$

1. Flip a random coin $c \leftarrow \{0, 1\}$.
2. If $D(y, c) = 1$ output c , otherwise, output \bar{c} .

OWP to PRG cont.

- ▶ $\Pr[D(f(U_n), b(U_n)) = 1] = \delta + \varepsilon'$
- ▶ $\Pr[D(f(U_n), \overline{b(U_n)}) = 1] = \delta - \varepsilon'$

Algorithm 9 (P)

Input: $y \in \{0, 1\}^n$

1. Flip a random coin $c \leftarrow \{0, 1\}$.
2. If $D(y, c) = 1$ output c , otherwise, output \bar{c} .

- ▶ It follows that

$$\begin{aligned}\Pr[P(f(U_n)) = b(U_n)] &= \Pr[c = b(U_n)] \cdot \Pr[D(f(U_n), c) = 1 \mid c = b(U_n)] \\ &\quad + \Pr[c = \overline{b(U_n)}] \cdot \Pr[D(f(U_n), c) = 0 \mid c = \overline{b(U_n)}]\end{aligned}$$

OWP to PRG cont.

- ▶ $\Pr[D(f(U_n), b(U_n)) = 1] = \delta + \varepsilon'$
- ▶ $\Pr[D(f(U_n), \overline{b(U_n)}) = 1] = \delta - \varepsilon'$

Algorithm 9 (P)

Input: $y \in \{0, 1\}^n$

1. Flip a random coin $c \leftarrow \{0, 1\}$.
2. If $D(y, c) = 1$ output c , otherwise, output \bar{c} .

- ▶ It follows that

$$\begin{aligned}\Pr[P(f(U_n)) = b(U_n)] &= \Pr[c = b(U_n)] \cdot \Pr[D(f(U_n), c) = 1 \mid c = b(U_n)] \\ &\quad + \Pr[c = \overline{b(U_n)}] \cdot \Pr[D(f(U_n), c) = 0 \mid c = \overline{b(U_n)}] \\ &= \frac{1}{2} \cdot (\delta + \varepsilon') + \frac{1}{2}(1 - \delta + \varepsilon') = \frac{1}{2} + \varepsilon'.\end{aligned}$$

Part IV

PRG from Regular OWF

Computational notions of entropy

Definition 10

X has (s, ϵ) -pseudoentropy at least k , if \exists rv Y with $H(Y) \geq k$ and X, Y are (s, ϵ) -indistinguishable.

(s, ϵ) -pseudo min/Reiny -entropy are analogously defined.

Computational notions of entropy

Definition 10

X has (s, ϵ) -pseudoentropy at least k , if \exists rv Y with $H(Y) \geq k$ and X, Y are (s, ϵ) -indistinguishable.

(s, ϵ) -pseudo min/Reiny -entropy are analogously defined.

► Example

Computational notions of entropy

Definition 10

X has (s, ϵ) -pseudoentropy at least k , if \exists rv Y with $H(Y) \geq k$ and X, Y are (s, ϵ) -indistinguishable.

(s, ϵ) -pseudo min/Reiny -entropy are analogously defined.

- ▶ Example
- ▶ Repeated sampling

Computational notions of entropy

Definition 10

X has (s, ϵ) -pseudoentropy at least k , if \exists rv Y with $H(Y) \geq k$ and X, Y are (s, ϵ) -indistinguishable.

(s, ϵ) -pseudo min/Reiny -entropy are analogously defined.

- ▶ Example
- ▶ Repeated sampling
- ▶ Non-monotonicity

Computational notions of entropy

Definition 10

X has (s, ϵ) -pseudoentropy at least k , if \exists rv Y with $H(Y) \geq k$ and X, Y are (s, ϵ) -indistinguishable.

(s, ϵ) -pseudo min/Reiny -entropy are analogously defined.

- ▶ Example
- ▶ Repeated sampling
- ▶ Non-monotonicity
- ▶ Ensembles

Computational notions of entropy

Definition 10

X has (s, ϵ) -pseudoentropy at least k , if \exists rv Y with $H(Y) \geq k$ and X, Y are (s, ϵ) -indistinguishable.

(s, ϵ) -pseudo min/Reiny -entropy are analogously defined.

- ▶ Example
- ▶ Repeated sampling
- ▶ Non-monotonicity
- ▶ Ensembles
- ▶ In the following we will simply write (s, ϵ) -entropy, etc

High entropy OWF from regular OWF

Claim 11

Let $f: \{0, 1\}^n \mapsto \{0, 1\}^n$ be a 2^k -regular (s, ε) -one-way, let $\mathcal{H} = \{h: \{0, 1\}^n \mapsto \{0, 1\}^{k+2}\}$ be 2-universal family, and let $g(h, x) = (f(x), h, h(x))$. Then

1. $H_2(g(U_n, H)) \geq 2n - \frac{1}{2}$, for $H \leftarrow \mathcal{H}$.
2. g is $(\Theta(s\varepsilon^2), 2\varepsilon)$ -one-way.

- ▶ k and m and \mathcal{H} are parameterized by n
- ▶ We assume $\log |\mathcal{H}| = n$ and $s \geq n$

g has high Renyi entropy

g has high Renyi entropy

$$\begin{aligned}\text{CP}(g(U_n, H)) &:= \Pr_{w, w' \leftarrow \{0,1\}^n \times \mathcal{H}} [g(w) = g(w')] \\ &= \Pr_{h, h' \leftarrow \mathcal{H}} [h = h'] \cdot \Pr_{(x, x') \leftarrow (\{0,1\}^n)^2} [f(x) = f(x')] \\ &\quad \cdot \Pr_{h \leftarrow \mathcal{H}; (x, x') \leftarrow (\{0,1\}^n)^2} [h(x) = h(x') \mid f(x) = f(x')]\end{aligned}$$

g has high Renyi entropy

$$\begin{aligned}\text{CP}(g(U_n, H)) &:= \Pr_{w, w' \leftarrow \{0,1\}^n \times \mathcal{H}} [g(w) = g(w')] \\ &= \Pr_{h, h' \leftarrow \mathcal{H}} [h = h'] \cdot \Pr_{(x, x') \leftarrow (\{0,1\}^n)^2} [f(x) = f(x')] \\ &\quad \cdot \Pr_{h \leftarrow \mathcal{H}; (x, x') \leftarrow (\{0,1\}^n)^2} [h(x) = h(x') \mid f(x) = f(x')] \\ &= \text{CP}(H) \cdot \text{CP}(f(U_n)) \cdot (2^{-k} + (1 - 2^{-k}) \cdot 2^{-k-2})\end{aligned}$$

g has high Renyi entropy

$$\begin{aligned}\text{CP}(g(U_n, H)) &:= \Pr_{w, w' \leftarrow \{0,1\}^n \times \mathcal{H}} [g(w) = g(w')] \\ &= \Pr_{h, h' \leftarrow \mathcal{H}} [h = h'] \cdot \Pr_{(x, x') \leftarrow (\{0,1\}^n)^2} [f(x) = f(x')] \\ &\quad \cdot \Pr_{h \leftarrow \mathcal{H}; (x, x') \leftarrow (\{0,1\}^n)^2} [h(x) = h(x') \mid f(x) = f(x')] \\ &= \text{CP}(H) \cdot \text{CP}(f(U_n)) \cdot (2^{-k} + (1 - 2^{-k}) \cdot 2^{-k-2}) \\ &\leq \text{CP}(H) \cdot \text{CP}(f(U_n)) \cdot 2^{-k} \cdot \frac{5}{4}\end{aligned}$$

g has high Renyi entropy

$$\begin{aligned}\text{CP}(g(U_n, H)) &:= \Pr_{w, w' \leftarrow \{0,1\}^n \times \mathcal{H}} [g(w) = g(w')] \\ &= \Pr_{h, h' \leftarrow \mathcal{H}} [h = h'] \cdot \Pr_{(x, x') \leftarrow (\{0,1\}^n)^2} [f(x) = f(x')] \\ &\quad \cdot \Pr_{h \leftarrow \mathcal{H}; (x, x') \leftarrow (\{0,1\}^n)^2} [h(x) = h(x') \mid f(x) = f(x')] \\ &= \text{CP}(H) \cdot \text{CP}(f(U_n)) \cdot (2^{-k} + (1 - 2^{-k}) \cdot 2^{-k-2}) \\ &\leq \text{CP}(H) \cdot \text{CP}(f(U_n)) \cdot 2^{-k} \cdot \frac{5}{4} = 2^{-n} \cdot 2^{-n} \cdot \frac{5}{4}.\end{aligned}$$

g has high Renyi entropy

$$\begin{aligned}\text{CP}(g(U_n, H)) &:= \Pr_{w, w' \leftarrow \{0,1\}^n \times \mathcal{H}} [g(w) = g(w')] \\ &= \Pr_{h, h' \leftarrow \mathcal{H}} [h = h'] \cdot \Pr_{(x, x') \leftarrow (\{0,1\}^n)^2} [f(x) = f(x')] \\ &\quad \cdot \Pr_{h \leftarrow \mathcal{H}; (x, x') \leftarrow (\{0,1\}^n)^2} [h(x) = h(x') \mid f(x) = f(x')] \\ &= \text{CP}(H) \cdot \text{CP}(f(U_n)) \cdot (2^{-k} + (1 - 2^{-k}) \cdot 2^{-k-2}) \\ &\leq \text{CP}(H) \cdot \text{CP}(f(U_n)) \cdot 2^{-k} \cdot \frac{5}{4} = 2^{-n} \cdot 2^{-n} \cdot \frac{5}{4}.\end{aligned}$$

g has high Renyi entropy

$$\begin{aligned}\text{CP}(g(U_n, H)) &:= \Pr_{w, w' \leftarrow \{0,1\}^n \times \mathcal{H}} [g(w) = g(w')] \\ &= \Pr_{h, h' \leftarrow \mathcal{H}} [h = h'] \cdot \Pr_{(x, x') \leftarrow (\{0,1\}^n)^2} [f(x) = f(x')] \\ &\quad \cdot \Pr_{h \leftarrow \mathcal{H}; (x, x') \leftarrow (\{0,1\}^n)^2} [h(x) = h(x') \mid f(x) = f(x')] \\ &= \text{CP}(H) \cdot \text{CP}(f(U_n)) \cdot (2^{-k} + (1 - 2^{-k}) \cdot 2^{-k-2}) \\ &\leq \text{CP}(H) \cdot \text{CP}(f(U_n)) \cdot 2^{-k} \cdot \frac{5}{4} = 2^{-n} \cdot 2^{-n} \cdot \frac{5}{4}.\end{aligned}$$

Hence, $H_2(g(U_n, H)) \geq 2n + \log \frac{4}{5} \geq 2n - \frac{1}{2}$.

g is one-way

g is one-way

Let A be an s' -size algorithm that inverts g w.p ϵ' and let $\ell = k - \lceil 2 \log \frac{1}{\epsilon'} \rceil$.

g is one-way

Let A be an s' -size algorithm that inverts g w.p ϵ' and let $\ell = k - \lceil 2 \log \frac{1}{\epsilon'} \rceil$.

Consider the following inverter for f

Algorithm 12 (B)

Input: $y \in \{0, 1\}^n$.

Return $D(y, h, z)$, for $h \leftarrow \mathcal{H}$ and $z \leftarrow \{0, 1\}^\ell$.

Algorithm 13 (D)

Input: $y \in \{0, 1\}^n$, $h \in \mathcal{H}$ and $z_1 \in \{0, 1\}^\ell$.

For all $z_2 \in \{0, 1\}^{k+2-\ell}$:

1. Let $(x, h) = A(y, h, z_1 \circ z_2)$.
2. If $f(x) = y$, return x .

g is one-way

Let A be an s' -size algorithm that inverts g w.p ϵ' and let $\ell = k - \lceil 2 \log \frac{1}{\epsilon'} \rceil$.

Consider the following inverter for f

Algorithm 12 (B)

Input: $y \in \{0, 1\}^n$.

Return $D(y, h, z)$, for $h \leftarrow \mathcal{H}$ and $z \leftarrow \{0, 1\}^\ell$.

Algorithm 13 (D)

Input: $y \in \{0, 1\}^n$, $h \in \mathcal{H}$ and $z_1 \in \{0, 1\}^\ell$.

For all $z_2 \in \{0, 1\}^{k+2-\ell}$:

1. Let $(x, h) = A(y, h, z_1 \circ z_2)$.
2. If $f(x) = y$, return x .

► B's size is $((s' + O(n)) \cdot 2^{2 \log \frac{1}{\epsilon'} + 2} = \Theta(s'/\epsilon^2)$

g is one-way

Let A be an s' -size algorithm that inverts g w.p ε' and let $\ell = k - \lceil 2 \log \frac{1}{\varepsilon'} \rceil$.

Consider the following inverter for f

Algorithm 12 (B)

Input: $y \in \{0, 1\}^n$.

Return $D(y, h, z)$, for $h \leftarrow \mathcal{H}$ and $z \leftarrow \{0, 1\}^\ell$.

Algorithm 13 (D)

Input: $y \in \{0, 1\}^n$, $h \in \mathcal{H}$ and $z_1 \in \{0, 1\}^\ell$.

For all $z_2 \in \{0, 1\}^{k+2-\ell}$:

1. Let $(x, h) = A(y, h, z_1 \circ z_2)$.
2. If $f(x) = y$, return x .

- ▶ B's size is $((s' + O(n)) \cdot 2^{2 \log \frac{1}{\varepsilon'} + 2} = \Theta(s'/\varepsilon^2)$
- ▶ $\Pr_{x \leftarrow \{0,1\}^n; h \leftarrow \mathcal{H}} [D(f(x), h, h(x)_{1,\dots,\ell}) \in f^{-1}(f(x))] = \varepsilon'$

g is one-way, cont.

We saw that

$$\Pr_{x \leftarrow \{0,1\}^n; h \leftarrow \mathcal{H}} [D(f(x), h, h(x)_1, \dots, \ell) \in f^{-1}(f(x))] = \varepsilon' \quad (1)$$

g is one-way, cont.

We saw that

$$\Pr_{x \leftarrow \{0,1\}^n; h \leftarrow \mathcal{H}} [D(f(x), h, h(x)_1, \dots, h(x)_\ell) \in f^{-1}(f(x))] = \varepsilon' \quad (1)$$

By the leftover hash lemma

$$\text{SD}((f(x), h, h(x)_1, \dots, h(x)_\ell)_{x \leftarrow \{0,1\}, h \leftarrow \mathcal{H}}, (f(x), h, U_\ell)_{x \leftarrow \{0,1\}, h \leftarrow \mathcal{H}}) \leq \varepsilon'/2 \quad (2)$$

g is one-way, cont.

We saw that

$$\Pr_{x \leftarrow \{0,1\}^n; h \leftarrow \mathcal{H}} [D(f(x), h, h(x)_1, \dots, h(x)_\ell) \in f^{-1}(f(x))] = \varepsilon' \quad (1)$$

By the leftover hash lemma

$$\text{SD}((f(x), h, h(x)_1, \dots, h(x)_\ell)_{x \leftarrow \{0,1\}^n, h \leftarrow \mathcal{H}}, (f(x), h, U_\ell)_{x \leftarrow \{0,1\}^n, h \leftarrow \mathcal{H}}) \leq \varepsilon'/2 \quad (2)$$

Hence,

$$\Pr_{x \leftarrow \{0,1\}^n} [B(f(x)) \in f^{-1}(f(x))] \geq \varepsilon' - \varepsilon'/2 = \varepsilon'/2.$$

The generator

Claim 14

Let $g: \{0, 1\}^n \mapsto \{0, 1\}^m$ be a function with $H_2(g(U_n)) \geq n - \frac{1}{2}$, and let b be (s, ϵ) -hardcore predicate for g . Then $v(U_n) = (g(U_n), b(U_n))$ has (s, ϵ) -Renyi-entropy $n + \frac{1}{2}$.

The generator

Claim 14

Let $g: \{0, 1\}^n \mapsto \{0, 1\}^m$ be a function with $H_2(g(U_n)) \geq n - \frac{1}{2}$, and let b be (s, ϵ) -hardcore predicate for g . Then $v(U_n) = (g(U_n), b(U_n))$ has (s, ϵ) -Renyi-entropy $n + \frac{1}{2}$.

Proof: ?

The generator

Claim 14

Let $g: \{0, 1\}^n \mapsto \{0, 1\}^m$ be a function with $H_2(g(U_n)) \geq n - \frac{1}{2}$, and let b be (s, ϵ) -hardcore predicate for g . Then $v(U_n) = (g(U_n), b(U_n))$ has (s, ϵ) -Renyi-entropy $n + \frac{1}{2}$.

Proof: ?

We call such v a **pseudo Renyi-entropy** generator.

The generator

Claim 14

Let $g: \{0, 1\}^n \mapsto \{0, 1\}^m$ be a function with $H_2(g(U_n)) \geq n - \frac{1}{2}$, and let b be (s, ε) -hardcore predicate for g . Then $v(U_n) = (g(U_n), b(U_n))$ has (s, ε) -Renyi-entropy $n + \frac{1}{2}$.

Proof: ?

We call such v a **pseudo Renyi-entropy** generator.

Claim 15

The function $v^n(x_1, \dots, x_n) = (v(x_1), \dots, v(x_n))$ has $(s - n^2, n\varepsilon)$ -Renyi-entropy $n^2 + \frac{n}{2}$.

The generator

Claim 14

Let $g: \{0, 1\}^n \mapsto \{0, 1\}^m$ be a function with $H_2(g(U_n)) \geq n - \frac{1}{2}$, and let b be (s, ε) -hardcore predicate for g . Then $v(U_n) = (g(U_n), b(U_n))$ has (s, ε) -Renyi-entropy $n + \frac{1}{2}$.

Proof: ?

We call such v a **pseudo Renyi-entropy** generator.

Claim 15

The function $v^n(x_1, \dots, x_n) = (v(x_1), \dots, v(x_n))$ has $(s - n^2, n\varepsilon)$ -Renyi-entropy $n^2 + \frac{n}{2}$.

Proof:

The generator

Claim 14

Let $g: \{0, 1\}^n \mapsto \{0, 1\}^m$ be a function with $H_2(g(U_n)) \geq n - \frac{1}{2}$, and let b be (s, ε) -hardcore predicate for g . Then $v(U_n) = (g(U_n), b(U_n))$ has (s, ε) -Renyi-entropy $n + \frac{1}{2}$.

Proof: ?

We call such v a **pseudo Renyi-entropy** generator.

Claim 15

The function $v^n(x_1, \dots, x_n) = (v(x_1), \dots, v(x_n))$ has $(s - n^2, n\varepsilon)$ -Renyi-entropy $n^2 + \frac{n}{2}$.

Proof:

- ▶ Let Z be a rv with $H_2(Z) \geq n + \frac{1}{2}$ such that Z and $v(U_n)$ are (s, ε) indistinguishable.

The generator

Claim 14

Let $g: \{0, 1\}^n \mapsto \{0, 1\}^m$ be a function with $H_2(g(U_n)) \geq n - \frac{1}{2}$, and let b be (s, ε) -hardcore predicate for g . Then $v(U_n) = (g(U_n), b(U_n))$ has (s, ε) -Renyi-entropy $n + \frac{1}{2}$.

Proof: ?

We call such v a **pseudo Renyi-entropy** generator.

Claim 15

The function $v^n(x_1, \dots, x_n) = (v(x_1), \dots, v(x_n))$ has $(s - n^2, n\varepsilon)$ -Renyi-entropy $n^2 + \frac{n}{2}$.

Proof:

- ▶ Let Z be a rv with $H_2(Z) \geq n + \frac{1}{2}$ such that Z and $v(U_n)$ are (s, ε) indistinguishable.
- ▶ $H_2(Z^n) \geq n^2 + \frac{n}{2}$

The generator

Claim 14

Let $g: \{0, 1\}^n \mapsto \{0, 1\}^m$ be a function with $H_2(g(U_n)) \geq n - \frac{1}{2}$, and let b be (s, ϵ) -hardcore predicate for g . Then $v(U_n) = (g(U_n), b(U_n))$ has (s, ϵ) -Renyi-entropy $n + \frac{1}{2}$.

Proof: ?

We call such v a **pseudo Renyi-entropy** generator.

Claim 15

The function $v^n(x_1, \dots, x_n) = (v(x_1), \dots, v(x_n))$ has $(s - n^2, n\epsilon)$ -Renyi-entropy $n^2 + \frac{n}{2}$.

Proof:

- ▶ Let Z be a rv with $H_2(Z) \geq n + \frac{1}{2}$ such that Z and $v(U_n)$ are (s, ϵ) indistinguishable.
- ▶ $H_2(Z^n) \geq n^2 + \frac{n}{2}$
- ▶ Z^n and $v^n(U_n)$ are $(s - n^2, n\epsilon)$ indistinguishable

The generator

Claim 14

Let $g: \{0, 1\}^n \mapsto \{0, 1\}^m$ be a function with $H_2(g(U_n)) \geq n - \frac{1}{2}$, and let b be (s, ϵ) -hardcore predicate for g . Then $v(U_n) = (g(U_n), b(U_n))$ has (s, ϵ) -Renyi-entropy $n + \frac{1}{2}$.

Proof: ?

We call such v a **pseudo Renyi-entropy** generator.

Claim 15

The function $v^n(x_1, \dots, x_n) = (v(x_1), \dots, v(x_n))$ has $(s - n^2, n\epsilon)$ -Renyi-entropy $n^2 + \frac{n}{2}$.

Proof:

- ▶ Let Z be a rv with $H_2(Z) \geq n + \frac{1}{2}$ such that Z and $v(U_n)$ are (s, ϵ) indistinguishable.
- ▶ $H_2(Z^n) \geq n^2 + \frac{n}{2}$
- ▶ Z^n and $v^n(U_n)$ are $(s - n^2, n\epsilon)$ indistinguishable

The generator

Claim 14

Let $g: \{0, 1\}^n \mapsto \{0, 1\}^m$ be a function with $H_2(g(U_n)) \geq n - \frac{1}{2}$, and let b be (s, ϵ) -hardcore predicate for g . Then $v(U_n) = (g(U_n), b(U_n))$ has (s, ϵ) -Renyi-entropy $n + \frac{1}{2}$.

Proof: ?

We call such v a **pseudo Renyi-entropy** generator.

Claim 15

The function $v^n(x_1, \dots, x_n) = (v(x_1), \dots, v(x_n))$ has $(s - n^2, n\epsilon)$ -Renyi-entropy $n^2 + \frac{n}{2}$.

Proof:

- ▶ Let Z be a rv with $H_2(Z) \geq n + \frac{1}{2}$ such that Z and $v(U_n)$ are (s, ϵ) indistinguishable.
- ▶ $H_2(Z^n) \geq n^2 + \frac{n}{2}$
- ▶ Z^n and $v^n(U_n)$ are $(s - n^2, n\epsilon)$ indistinguishable

The generator cont.

The generator cont.

Claim 16

Let $\mathcal{H}: \{0, 1\}^{n^2+n} \mapsto \{0, 1\}^{n^2+n/4}$ be an 2-universal family and let $G: \{0, 1\}^n \times \mathcal{H}$ defined by $G(x_1, \dots, x_n, h) = (h, h(v^n(x_1, \dots, x_n)))$. Then $G(H, U_n^n)$ is $(s - n^2 - s_{\mathcal{H}}, n\varepsilon + 2^{-n/4})$ indistinguishable from $(H, U_{n^2+n/4})$, for $H \leftarrow \mathcal{H}$ and $s_{\mathcal{H}}$ being the size of sampling and evaluating algorithm for \mathcal{H} .

The generator cont.

Claim 16

Let $\mathcal{H}: \{0, 1\}^{n^2+n} \mapsto \{0, 1\}^{n^2+n/4}$ be an 2-universal family and let $G: \{0, 1\}^n \times \mathcal{H}$ defined by $G(x_1, \dots, x_n, h) = (h, h(v^n(x_1, \dots, x_n)))$. Then $G(H, U_n^n)$ is $(s - n^2 - s_{\mathcal{H}}, n\varepsilon + 2^{-n/4})$ indistinguishable from $(H, U_{n^2+n/4})$, for $H \leftarrow \mathcal{H}$ and $s_{\mathcal{H}}$ being the size of sampling and evaluating algorithm for \mathcal{H} .

Corollary 17

If f and b and \mathcal{H} (?) are poly-time computable, then G is a $(s - n^2 - s_{\mathcal{H}}, n\varepsilon + 2^{-n/4})$ -PRG.

The generator cont.

Claim 16

Let $\mathcal{H}: \{0, 1\}^{n^2+n} \mapsto \{0, 1\}^{n^2+n/4}$ be an 2-universal family and let $G: \{0, 1\}^n \times \mathcal{H}$ defined by $G(x_1, \dots, x_n, h) = (h, h(v^n(x_1, \dots, x_n)))$. Then $G(H, U_n^n)$ is $(s - n^2 - s_{\mathcal{H}}, n\varepsilon + 2^{-n/4})$ indistinguishable from $(H, U_{n^2+n/4})$, for $H \leftarrow \mathcal{H}$ and $s_{\mathcal{H}}$ being the size of sampling and evaluating algorithm for \mathcal{H} .

Corollary 17

If f and b and \mathcal{H} (?) are poly-time computable, then G is a $(s - n^2 - s_{\mathcal{H}}, n\varepsilon + 2^{-n/4})$ -PRG.

Proof: (of claim)

The generator cont.

Claim 16

Let $\mathcal{H}: \{0, 1\}^{n^2+n} \mapsto \{0, 1\}^{n^2+n/4}$ be an 2-universal family and let $G: \{0, 1\}^n \times \mathcal{H}$ defined by $G(x_1, \dots, x_n, h) = (h, h(v^n(x_1, \dots, x_n)))$. Then $G(H, U_n^n)$ is $(s - n^2 - s_{\mathcal{H}}, n\varepsilon + 2^{-n/4})$ indistinguishable from $(H, U_{n^2+n/4})$, for $H \leftarrow \mathcal{H}$ and $s_{\mathcal{H}}$ being the size of sampling and evaluating algorithm for \mathcal{H} .

Corollary 17

If f and b and \mathcal{H} (?) are poly-time computable, then G is a $(s - n^2 - s_{\mathcal{H}}, n\varepsilon + 2^{-n/4})$ -PRG.

Proof: (of claim) Let Z with $H_2(Z) \geq n + \frac{1}{2}$ and $Z, v(U_n)$ are (s, ε) -indist.

The generator cont.

Claim 16

Let $\mathcal{H}: \{0, 1\}^{n^2+n} \mapsto \{0, 1\}^{n^2+n/4}$ be an 2-universal family and let $G: \{0, 1\}^n \times \mathcal{H}$ defined by $G(x_1, \dots, x_n, h) = (h, h(v^n(x_1, \dots, x_n)))$. Then $G(H, U_n^n)$ is $(s - n^2 - s_{\mathcal{H}}, n\varepsilon + 2^{-n/4})$ indistinguishable from $(H, U_{n^2+n/4})$, for $H \leftarrow \mathcal{H}$ and $s_{\mathcal{H}}$ being the size of sampling and evaluating algorithm for \mathcal{H} .

Corollary 17

If f and b and \mathcal{H} (?) are poly-time computable, then G is a $(s - n^2 - s_{\mathcal{H}}, n\varepsilon + 2^{-n/4})$ -PRG.

Proof: (of claim) Let Z with $H_2(Z) \geq n + \frac{1}{2}$ and $Z, v(U_n)$ are (s, ε) -indist.

- By the leftover hash lemma $SD((H, H(Z^n)), (H, U_{n^2+n/4})) \leq 2^{-n/4}$

The generator cont.

Claim 16

Let $\mathcal{H}: \{0, 1\}^{n^2+n} \mapsto \{0, 1\}^{n^2+n/4}$ be an 2-universal family and let $G: \{0, 1\}^n \times \mathcal{H}$ defined by $G(x_1, \dots, x_n, h) = (h, h(v^n(x_1, \dots, x_n)))$. Then $G(H, U_n^n)$ is $(s - n^2 - s_{\mathcal{H}}, n\varepsilon + 2^{-n/4})$ indistinguishable from $(H, U_{n^2+n/4})$, for $H \leftarrow \mathcal{H}$ and $s_{\mathcal{H}}$ being the size of sampling and evaluating algorithm for \mathcal{H} .

Corollary 17

If f and b and \mathcal{H} (?) are poly-time computable, then G is a $(s - n^2 - s_{\mathcal{H}}, n\varepsilon + 2^{-n/4})$ -PRG.

Proof: (of claim) Let Z with $H_2(Z) \geq n + \frac{1}{2}$ and $Z, v(U_n)$ are (s, ε) -indist.

- ▶ By the leftover hash lemma $SD((H, H(Z^n)), (H, U_{n^2+n/4})) \leq 2^{-n/4}$
- ▶ Let D be an s' -size algorithm that distinguishes $G(U_n^n, H)$ from $(H, U_{n^2+n/4})$ with advantage $\varepsilon' + 2^{-n/4}$

The generator cont.

Claim 16

Let $\mathcal{H}: \{0, 1\}^{n^2+n} \mapsto \{0, 1\}^{n^2+n/4}$ be an 2-universal family and let $G: \{0, 1\}^n \times \mathcal{H}$ defined by $G(x_1, \dots, x_n, h) = (h, h(v^n(x_1, \dots, x_n)))$. Then $G(H, U_n^n)$ is $(s - n^2 - s_{\mathcal{H}}, n\varepsilon + 2^{-n/4})$ indistinguishable from $(H, U_{n^2+n/4})$, for $H \leftarrow \mathcal{H}$ and $s_{\mathcal{H}}$ being the size of sampling and evaluating algorithm for \mathcal{H} .

Corollary 17

If f and b and \mathcal{H} (?) are poly-time computable, then G is a $(s - n^2 - s_{\mathcal{H}}, n\varepsilon + 2^{-n/4})$ -PRG.

Proof: (of claim) Let Z with $H_2(Z) \geq n + \frac{1}{2}$ and $Z, v(U_n)$ are (s, ε) -indist.

- ▶ By the leftover hash lemma $SD((H, H(Z^n)), (H, U_{n^2+n/4})) \leq 2^{-n/4}$
- ▶ Let D be an s' -size algorithm that distinguishes $G(U_n^n, H)$ from $(H, U_{n^2+n/4})$ with advantage $\varepsilon' + 2^{-n/4}$
- ▶ Hence, $\exists (s' + s_{\mathcal{H}})$ -size algorithm that distinguishes $v^n(U_n^n)$ from Z^n with advantage ε'

The generator cont.

Claim 16

Let $\mathcal{H}: \{0, 1\}^{n^2+n} \mapsto \{0, 1\}^{n^2+n/4}$ be an 2-universal family and let $G: \{0, 1\}^n \times \mathcal{H}$ defined by $G(x_1, \dots, x_n, h) = (h, h(v^n(x_1, \dots, x_n)))$. Then $G(H, U_n^n)$ is $(s - n^2 - s_{\mathcal{H}}, n\varepsilon + 2^{-n/4})$ indistinguishable from $(H, U_{n^2+n/4})$, for $H \leftarrow \mathcal{H}$ and $s_{\mathcal{H}}$ being the size of sampling and evaluating algorithm for \mathcal{H} .

Corollary 17

If f and b and \mathcal{H} (?) are poly-time computable, then G is a $(s - n^2 - s_{\mathcal{H}}, n\varepsilon + 2^{-n/4})$ -PRG.

Proof: (of claim) Let Z with $H_2(Z) \geq n + \frac{1}{2}$ and $Z, v(U_n)$ are (s, ε) -indist.

- ▶ By the leftover hash lemma $SD((H, H(Z^n)), (H, U_{n^2+n/4})) \leq 2^{-n/4}$
- ▶ Let D be an s' -size algorithm that distinguishes $G(U_n^n, H)$ from $(H, U_{n^2+n/4})$ with advantage $\varepsilon' + 2^{-n/4}$
- ▶ Hence, $\exists (s' + s_{\mathcal{H}})$ -size algorithm that distinguishes $v^n(U_n^n)$ from Z^n with advantage ε'
- ▶ Hence $s' \leq s - n^2 - s_{\mathcal{H}} \implies \varepsilon' \leq n\varepsilon$.

Remarks

- ▶ PRG “length extension”

Remarks

- ▶ PRG “length extension”
- ▶ PRG from any OWF